A History of the Capability Maturity Model® for Software

Mark C. Paulk

mcp@sei.cmu.edu -or- Mark.Paulk@ieee.org
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA  15213-3890


Topics

Setting Context: the “Prehistory” of the CMM

“Versions” of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

Closing Comments
What Is the Software CMM?

A common-sense application of process management and quality improvement concepts to software development and maintenance

A community-developed guide

A model for organizational improvement

The underlying structure for reliable and consistent CMM-based appraisal methods

Software CMM v1.1 Key Process Areas

<table>
<thead>
<tr>
<th>Level</th>
<th>Focus</th>
<th>Key Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Optimizing</td>
<td>Continuous process improvement, Defect Prevention, Technology Change Management,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Process Change Management</td>
</tr>
<tr>
<td>4</td>
<td>Managed</td>
<td>Product and process quality, Quantitative Process Management, Software Quality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Management</td>
</tr>
<tr>
<td>3</td>
<td>Defined</td>
<td>Engineering processes and organizational support, Organization Process Focus,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Organization Process Definition, Training Program, Integrated Software Management,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Product Engineering, Intergroup Coordination, Peer Reviews</td>
</tr>
<tr>
<td>2</td>
<td>Repeatable</td>
<td>Project management processes, Requirements Management, Software Project Planning,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Project Tracking &amp; Oversight, Software Subcontract Management, Software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quality Assurance, Software Configuration Management</td>
</tr>
<tr>
<td>1</td>
<td>Initial</td>
<td>Competent people (and heroics)</td>
</tr>
</tbody>
</table>
A History of the Software CMM

The history of the CMM can be summarized by the different incarnations it has gone through.
• several challenging decisions were made at different points in CMM development
• some issues were identified that remain problematic
• “issue slides” were retrieved from the archives

A fuller picture includes some alternate approaches that were unsuccessful and why.
• software process domains
• normative model

Inspirations for the Software CMM

Dissatisfaction with known, consistent software problems

Total Quality Management (TQM) successes

Crosby’s maturity grid

IBM’s process grid
The State of the Practice

“I’d rather have it wrong than have it late. We can always fix it later.”
- A senior software manager (industry)

“The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost.”
- A program manager (government)

Or see any shrinkwrap software warranty…

Evolution of Process Capability

<table>
<thead>
<tr>
<th>Level</th>
<th>Process Characteristics</th>
<th>Predicted Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Optimizing</td>
<td>Process improvement is institutionalized</td>
</tr>
<tr>
<td>4</td>
<td>Managed</td>
<td>Project and process are quantitatively controlled</td>
</tr>
<tr>
<td>3</td>
<td>Defined</td>
<td>Software engineering and management processes defined and integrated</td>
</tr>
<tr>
<td>2</td>
<td>Repeatable</td>
<td>Project management system in place; performance is repeatable</td>
</tr>
<tr>
<td>1</td>
<td>Initial</td>
<td>Process is informal and unpredictable</td>
</tr>
</tbody>
</table>
AFIT Study


Impact of Software Process Improvement: Boeing Data

John Vu, Boeing, keynote talk at SEPG '97, "Software Process Improvement Journey (From Level 1 to Level 5)"
“Trends” in Quality Results

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Design Faults / KSLOC (Keene)</th>
<th>Delivered Defects / FP (Jones)</th>
<th>Shipped Defects / KSLOC (Krasner)</th>
<th>Relative Defect Density (Williams)</th>
<th>Shipped Defects (Rifkin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.05</td>
<td>0.5</td>
<td>0.05</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.14</td>
<td>2.5</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.27</td>
<td>3.5</td>
<td>0.2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.44</td>
<td>6</td>
<td>0.4</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>5-6</td>
<td>0.75</td>
<td>30</td>
<td>1.0</td>
<td>61</td>
</tr>
</tbody>
</table>

Karl D. Williams, "The Value of Software Improvement... Results! Results! Results!" SPIRE97, 4 June 1997.

Impact on Effort

In COCOMO II, the PMAT variable factors in maturity level in terms of decreasing effort/cost.
- one level change results in 15-21% decrease in effort


Myth: Software Problems Are "Technical" Problems

Examined real-life case studies
- "Bugs in the Program" report, 1989
- red teams, assessments, evaluations, …

Well-known, consistent problems – revealing a major gap between the state-of-the-art and the state-of-the-practice

The major problems in software development are managerial – not technical.

Definition of Software Process

Process – a sequence of steps performed for a given purpose (IEEE)

Software process – a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (SEI)
Process Management Premise

The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it.

This premise implies focus on process as well as product.

The value of this premise is visible world-wide in the Total Quality Management movements in the manufacturing and service industries.

Applying Total Quality Management to Software

Process improvement fits in an overall business context — CMM applies to software.
Crosby


Quality is measured by the cost of quality which is the expense of nonconformance - the cost of doing things wrong.

**Stages**
- Stage 1: Uncertainty
- Stage 2: Awakening
- Stage 3: Enlightenment
- Stage 4: Wisdom
- Stage 5: Certainty

**Measurement Categories**
- management understanding and attitude
- quality organization status
- problem handling
- cost of quality as percent of sales
- quality improvement actions
- summation of company quality posture

---

**IBM Maturity Grid**


**Five-Point Scale**
- Traditional
- Awareness
- Knowledge
- Skill & wisdom
- Integrated management system

**Attributes**
- process
- methods
- adherence to practices
- tools
- change control
- data gathering
- data communication and use
- goal setting
- quality focus
- customer focus
- technical awareness

**Process Stages**
- requirements
- product level design
- component level design
- module level design
- code
- unit test
- functional verification test
- product verification test
- system verification test
- package and release
- early support program
- general availability
1987: Characterizing the Software Process


Software Process Maturity Framework

- **Initial (1)**
  - Unpredictable and poorly controlled

- **Repeatable (2)**
  - Can repeat previously mastered tasks

- **Defined (3)**
  - Process characterized, fairly well understood

- **Managed (4)**
  - Process measured and controlled

- **Optimizing (5)**
  - Focus on process improvement
**Key Actions in 1987**

- **Initial (Level 1)**
- **Repeatable (Level 2)**
  - Project Management
  - Management Oversight
  - Product Assurance
  - Change Control of Requirements
- **Defined (Level 3)**
  - Process Group
  - Process Architecture
  - Software Engineering Methods
- **Managed (Level 4)**
  - Process Measurement
  - Process Database
  - Process Analysis
  - Product Quality
- **Optimized (Level 5)**
  - Automated Support
  - Process Optimization

---

**Issue: Questionnaire as Model**

The 1987 maturity questionnaire was widely viewed as the model.
- 85 process questions
- 16 technology stage questions

Technology stage questions were effectively never used.

Appraisal based solely on questionnaire would be superficial.
- neither assessments nor appraisals ever relied solely on the questionnaire (as trained)
**Issue: Level of Abstraction**

The 1987 software process maturity framework was abstract and incomplete.

Difficult to explain why questions were at a particular maturity level.

Significant interpretation issues existed.
- “technical reviews” in the questionnaire versus “inspections”
- etc.

**Issue: Scoring**

If the maturity questionnaire is not the sole basis for scoring, what is the algorithm?

Essentially relied on finding problems, then mapping to related questions as necessary to adjust score.
- if questions are good probes, there will be some related to any problems
**Issue: Reliability & Consistency**

Appraisals were based on an “expert judgment” approach.

Danger of inconsistent results between teams (e.g., two-level differences for the same organization).

Danger of unreliable appraisals by the same team, based on normal human variation.

---

**1988: SCE Team Training**

In 1988 the SEI developed training courses for software capability evaluation teams.

A two-hour module in the SCE team training described the key challenges in moving between levels.

Concern about reliability and consistency of evaluations drove the inclusion of this module.
- no equivalent in software process assessment training
Key Challenges in 1988

- Optimizing (Level 5)
  - Problem Prevention
  - Problem Analysis
  - Changing Technology
- Managed (Level 4)
  - Process Measurement
  - Process Analysis
  - Quantitative Quality Plans
- Defined (Level 3)
  - Process Groups
  - Standards
  - Training
  - Testing
  - Reviews
- Repeatable (Level 2)
  - Change Control
  - Project Planning
  - Project Management
  - Software Quality Assurance
- Initial (Level 1)

Key: Degree of Change
- Red italics = major / conceptual
- Green italics = medium / content
- Black = relatively minor

Issue: Level of Detail

SCE training on the maturity model was only two hours.
- still superficial coverage
- still relied heavily on expert judgment
**Issue: Availability of Training**

For all practical purposes, SCE training was restricted to government employees performing evaluations.

- limited number of classes
- training materials not available outside class

---

**1989: Managing the Software Process**

Alternative Approaches

Two alternative approaches were attempted for formalizing the software process maturity framework
- software process domains
- normative model

Both approaches were part-time efforts within the Contractor Software Engineering Capability Assessment (CSECA) project (later SCE project).

Both approaches were abandoned after prototypes were reviewed within the program and/or with external reviewers.
July 1988: Software Process Domains

Planning
Measurement and analysis
Process definition
Verification and validation
Subcontract management
Software management
Technology insertion
Software engineering methods
Training
Selection and retention (of staff)

Issues With Domains

Short domain descriptions, approximately 1-2 pages apiece ⇒ still ambiguous

Domains spanned maturity levels ⇒ complex and ambiguous

Interdependencies between different domains, especially between practices at different maturity levels in different domains ⇒ complex and ambiguous
1989: Normative Model

“Orthogonal” representation of maturity in terms of stability factors and maturity indices applied to unit operations.

Stability Factors
- resources
- training
- tools
- plans
- policies
- responsibility

Maturity Indices
- existence
- review
- selection
- metrics
- analysis
- monitoring

Unit Operations
- staffing
- committing
- planning
- tracking
- executing
- documenting
- verifying

Issue: Usability

While the normative model led to some significant insights affecting the later development of the CMM, it was very difficult to explain.

It was deemed too artificial, or mathematically formal, to be useful for technology transition.

It failed the “usability” test (not easily comprehensible).
January 1990: CMM v0.2

The first draft of the Software CMM distributed for review outside the SEI.

Watts Humphrey was the Program Director for the Software Process Program but had identified Bill Curtis as his replacement.
Key Process Areas

Identify a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability.

Focused on a single maturity level, but practices could span maturity levels (especially subpractices).

Identify the issues that must be addressed to achieve a maturity level.

“Key Process Areas” in v0.2

- **Repeatable (Level 2)**
  - Project Planning
  - Project Management
  - Subcontract Management
  - Software Quality Assurance
  - Software Configuration Management

- **Defined (Level 3)**
  - Software Requirements Analysis
  - Software Design
  - Software Testing
  - Software Engineering Interfaces
  - Peer Reviews
  - Organizational Process Focus
  - Training Program

- **Managed (Level 4)**
  - Quality Management
  - Process Measurement and Analysis

- **Optimizing (Level 5)**
  - Process Improvement
  - Defect Prevention
  - (Contracting for Software dropped)
**Issue: KPAs Spanning Levels**

Should key process areas span maturity levels?

Feedback that (sub)practices at different levels were confusing.
- difficult to understand, implement, and transition
- focus on moving up maturity levels was obscured

Encouraged debates over why a practice was at a particular maturity level, debate the precedence of KPAs.

*Note that this observation (and this slide) preceded the SEI's work with ISO on continuous architectures and CMMI.*

---

**June 1990: CMM v0.6**

One of the later drafts of the Software CMM.

Bill Curtis had become Program Director for the Software Process Program.
“In earlier drafts of the CMM, key process areas spanned maturity levels but they did not span all maturity levels… The decision was made with version 0.6 to redefine the key process areas as residing at a single maturity level. As a result of defining key process areas as residing at a single maturity level, Software Project Management [later renamed Integrated Software Management] was added at level 3 to address software project planning and management issues at Maturity Level 3. This has been one of the more controversial decisions in defining the structure of the CMM… If key process areas span levels, then a more complete picture is provided, but the ‘vital few’ issues that dominate organizational improvement may be lost in the detail. Also, the emphasis on organization improvement in the CMM means that some processes are prioritized before others. When providing a detailed set of guidelines for process improvement, such as the key practices in the CMM, this unevenness in (described) process capability at different maturity levels becomes more visible and distracting than when the emphasis is explicitly placed on the ‘key’ areas that build organizational capability. Both perspectives are valuable, but the CMM’s explicit target is organizations.”

1991: CMM v1.0


**Issue: Scope**

What should the scope of the CMM be?

How far should we go into
- people issues?
- technology issues?
- concurrent engineering?
- organization culture/change?

Many TQM issues not addressed in CMM as scoping decision (software-specific)
- issues that are important for effective change and improvement

**Issue: Requirements Elicitation**

Whose job is it to identify what the software system should do?

Is requirements elicitation a systems engineering problem?

Which definition of quality will we use?
- conformance to requirements?
- customer satisfaction?
- customer delight?
Issue: An Audit Checklist

Is the CMM too detailed?

There is an enormous amount of information in this 500-page document.

Danger of CMM being used as an audit checklist.

Danger of CMM being inappropriately used in environments different than that for which the key practices were written.

1993: Software CMM v1.1

CMM v1.1 was a “minor” revision of v1.0 ...

There were name changes of KPAs and common features, but the content did not change significantly.

Almost every practice in the CMM was revised to improve consistency and clarity.
**Issue: The Customer’s Maturity**

Can a Level 1 acquisition agency effectively manage a Level 5 software supplier?

Without crippling the supplier’s software process?
Issue: Maintenance

Approximately 75% of software life cycle costs are in maintenance (sustaining engineering).

The CMM addresses the maintenance environment when appropriately interpreted.
• focus of practices is on the development environment
• as part of v1.1 revision, some practices were rewritten to make them more friendly to the maintenance environment

Issue: Appraiser Qualifications

What should the requirements be for becoming a licensed CMM-based appraiser?

Knowledge of the CMM?

Taking a CMM course (i.e., the Introduction to CMM)?

Auditing, interviewing, and reading skills?
Issue: Focusing on Level Number

Some organizations focus on what their maturity level is.
• senior executives may legitimately be interested in only the level of abstraction represented by the level
• most managers should be more interested in the key process area profile and the problems represented thereby

Danger is focusing on a score rather than on software process improvement.

Topics
Setting Context: the “Prehistory” of the CMM
“Versions” of the Software CMM

Related Work: ISO Standards
Software CMM v2
CMM Integration
Concluding Comments
Related Efforts – the “Quagmire”

“Children” of CMM / SPA / SCE
- Trillium
- Software Technology Diagnostic
- Healthcheck
- etc.

Government-procurement methods
- SDC/CR
- SDCE
- etc.

ISO standards – 9001, 15504, 12207,… 15288

CMM Gaps Identified in ISO 9001

4.7 Purchaser-Supplied Product
- purchaser-supplied and commercial-off-the-shelf (COTS) software addressed only in context of planning (ISM.AC.6.3) – use SSM

4.15 Handling, Storage, Packaging, and Delivery
- acceptance testing (SPE.AC.7) and release (SCM.AC.7) addressed
- installation process – including handling, storage, packaging, and delivery – not specifically addressed

4.19 Servicing
- ISO 9000-3 interprets servicing as maintenance (sustaining engineering)
- maintenance not separate component in CMM
International Standards Organization (ISO), 1991 Plenary

Proposal from Alec Dorling of the U.K. for a study period to examine whether consensus existed for creating an international standard on software process assessment

WG10 eventually established to build the standard

SPICE (Software Process Improvement and Capability dEtermination) is the name of the development and trialling group

Original SPICE Project Schedule

Developing drafts of standard components
  • 2Q 1993 through 4Q 1994

Key components of standard developed first
  • Baseline Practices Guide (BPG is reference model) and Process Assessment Guide completed 2Q 1994

Release to WG10 in 4Q 1994 *(done in 1995)*
  • 2 months of reviews
  • 20+ months of trials (SPICE product testing and revision)

Introducing Continuous Architecture

Two Architectural Perspectives

A “staged” architecture, e.g., the Software CMM
- focuses on building organizational capability
- identifies the vital few issues to focus on
- describes a roadmap for process improvement

The staged architecture was designed for changing organizational behavior.

A “continuous” architecture, e.g., ISO/IEC 15504
- focuses on building process capability
- provides a reference model for rating processes
- describes the terrain of process management

The continuous architecture was designed for comparing different models.
Topics

Setting Context: the “Prehistory” of the CMM

“Versions” of the Software CMM, 1987 to 1993

Related Work: ISO Standards

→ Software CMM v2

CMM Integration

Closing Comments

Drivers for Software CMM v2

Address change requests from users.

Continual improvement of the Software CMM
  • respond to growing/changing needs
  • improved understanding of “best practices”
  • improved understanding of maturity levels 4 and 5

Harmonize with relevant national and international standards (and other CMMs)
  • minimize unnecessary differences
  • provide mappings
Sources of Change

Change requests

Workshops
- February 1995 brainstorming workshop
- November 1995 requirements workshop
- April 1996 maturity level 4 & 5 workshop

Mappings to standards, other CMMs, working papers from community

Discussions with high maturity organizations

Pilots of prototypes

Software CMM Version 2

Balance between conflicting requirements
- stability of CMM for organizations engaged in software process improvement
- need to continually improve CMM to address user needs

Version 2 planned for release in November 1997
Integrating Other Maturity Models

A number of other models based on CMM being developed to address the specific needs of
• systems engineering (SE-CMM)
• people (PM-CMM)
• software acquisition (SA-CMM)
• discipline of engineering (engineering maturity model or EMM)

Both staged and continuous architecture models had been characterized as “CMMs.”

Model integration working group at SEI was studying model integration and generalization.

The Revision Process

Revision of the Software CMM involved
• collecting change requests
• identifying new practices and key process areas needed by users of the CMM
• distributing prototype key process areas for review
• pilot testing
• holding CMM workshops
• distributing draft versions of CMM v2 for review
• recommendations from CMM Advisory Board
• decisions by CMM Change Control Board
Software CMM v2 Release Halted

In October 1997, SEI’s sponsor, the Office of the Under Secretary of Defense for Acquisition and Technology, directed that the Software CMM Version 2 release be halted in favor of work on CMM Integration.

One of the source documents for CMMI is Software CMM v2C.

Topics

Setting Context: the “Prehistory” of the CMM

“Versions” of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

Closing Comments
CMMI Source Models

Industry
SEI
Government

• Team of Teams
• Modeling and Discipline Experts
• Collaborative Process

CMMI Product Suite

Model Comparisons

<table>
<thead>
<tr>
<th>Release</th>
<th>PAs/ FAs</th>
<th>Goals/ Themes*</th>
<th>Activities/ Practices**</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW-CMM V1.1</td>
<td>18</td>
<td>52</td>
<td>316</td>
</tr>
<tr>
<td>SW-CMM V2C</td>
<td>19</td>
<td>62</td>
<td>318</td>
</tr>
<tr>
<td>EIA/IS 731</td>
<td>19</td>
<td>77</td>
<td>383</td>
</tr>
<tr>
<td>IPD-CMM V0.98</td>
<td>23</td>
<td>60</td>
<td>865</td>
</tr>
<tr>
<td>CMMI V0.1 SE/SW</td>
<td>27</td>
<td>149</td>
<td>550</td>
</tr>
<tr>
<td>CMMI V0.2 SE/SW</td>
<td>24</td>
<td>80</td>
<td>528</td>
</tr>
<tr>
<td>CMMI V1.0 SE/SW</td>
<td>22</td>
<td>70</td>
<td>417</td>
</tr>
</tbody>
</table>

Legend:
* = Ratable components (Maturity Rating e.g., Level n)
** = Key to implementation effort
Comparing SW-CMM to CMMI
Staged Level 2 (Managed)

Comparing SW-CMM to CMMI
Staged Level 3 (Defined)
Comparing SW-CMM to CMMI
Staged Level 4 (Quantitatively Managed)

SW-CMM v1.1
Key Process Areas (KPAs)
- Quantitative Process Management
- Software Quality Management

CMMI
Process Areas (PAs)
- Organizational Process Performance
- Quantitative Project Management

Comparing SW-CMM to CMMI
Staged Level 5 (Optimizing)

SW-CMM v1.1
Key Process Areas (KPAs)
- Defect Prevention
- Technology Change Management
- Process Change Management

CMMI
Process Areas (PAs)
- Causal Analysis and Resolution
- Organizational Innovation and Deployment
The Stream of Continuous & Staged Architecture Models Leading to CMMI

1979 – Crosby’s maturity grid (Quality is Free)
1985 – IBM maturity grid (Radice)
1987 – SEI software process maturity framework
1988 – SEI software process domains
1989 – SEI normative model
1990 – SEI Software CMM v0.2
1990 – SEI Software CMM v0.6
1991 – SEI Software CMM v1.0
1993 – SEI Software CMM v1.1
1995 – Systems Engineering CMM
1997 – SEI Software CMM v2 Draft C
1998 – EIA 731 (Systems Engineering Capability Model)
1998 – ISO/IEC 15504 type 2 technical reports
2000 – SEI CMM Integration v1.0 (both)

Topics

Setting Context: the “Prehistory” of the CMM

“Versions” of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

Closing Comments
DoD Policy for CMMI

From Dr. Jack Ferguson, Director, Software Intensive Systems (OUSD (AT&L)), at Software Technology Conference, 2001

“CMMI will become the logical integrated successor for the CMM-SW for software engineering and EIA/IS 731 for systems engineering.”

“CMMI will become the approved means of judging engineering maturity for procurements within two years.”

SEI Policy for CMMI

From the CMMI FAQ, as of 17 September 2001

“The models that are designated as the starting point for the CMMI Product Suite development and identified as source documents will no longer be updated or supported by the issuing organization. The Product Suite is intended to replace the source models. As other disciplines are incorporated into the CMMI Product Suite, they too would follow the same process. As improvements are incorporated into the CMMI Product Suite, the original source documents will become obsolete and less representative of industry practice.

The plan is that such replacement would take place three years after successful model development and full release of the CMMI Product Suite. This replacement is currently scheduled for Fall 2003.”
General SEI Information

SEI Customer Relations  +1 (412) 268-5800

SEI FAX number  +1 (412) 268-5758

Internet Address
   customer-relations@sei.cmu.edu

Mailing Address
   Customer Relations
   Software Engineering Institute
   Carnegie Mellon University
   4500 Fifth Avenue
   Pittsburgh, PA  15213-3890

Internet Access to SEI

www.sei.cmu.edu
www.sei.cmu.edu/cmm/
www.sei.cmu.edu/cmm/cmm.articles.html
www.sei.cmu.edu/cmm/slides/slides.html
www.sei.cmu.edu/cmmi/

www.sei.cmu.edu/cmm/slides/cmm-history.pdf
   - www.sei.cmu.edu/cmm/slides/cmm-history-handout.pdf